Homework 1: Crypto Basics

This homework is due on **Thursday, February 13, 2020** at **11:59 p.m.** and counts for 8% of your course grade. Late submissions will be penalized by 10% plus an additional 10% every 5 hours until received. Late work will not be accepted after 24 hours past the deadline. If you have a conflict due to travel, interviews, etc., please plan accordingly and turn in your homework early.

Please submit one pdf file to CANVAS named **[Your PID]_HW1.pdf**. *Please do not submit an image of your handwritten solutions.* Make sure to comment your code with clear explanations and be sure to *cite any references used*.

We encourage you to discuss the problems and your general approach with other students in the class. However, the answers you turn in must be your own original work, and you are bound by the Honor Code. Submit your solution on CANVAS following the instructions at the end of this document.

To solve some of these problems, you might need to write some short programs; submit them with your answers. We recommend Python, but you may use any common language or numerical package.

1. **(10 points)** Here is some ciphertext that was produced by a simple substitution cipher that we learned about in class:

   ```
   OBRGXIMYAZZAWCATBNMUYYHAZNVGFCXPVVSIJSVLKIFAVGBIECAZSBWGRGRQWUCHMMOCYE
   FLGQQNKFSHQMGYALNKCIJQVEKVWXNFOYFYQBESGOYTXMAYTXSISNBPMSGOJBKFWRUTTMLS
   BNQMLLRGFNZUAWHZLBRVZGHUVZMCKJEHSLSWGXCNZYEXRIMLPXRIXNUXRNSNRPHFDHBMAY
   WKHTKNGNUXRNJUVGMYNYEYNLYYGPGYFSBNQQWUCHMMSLRWTFDYQRNOJUEWNLVUZIDHWXLH
   TLKNEXMALBRQGUMMGXCUFXLHTLLLRTROJYFIDHLIGADLUBVXENSCALVCDFFIQCFAHISILU
   XXZXNUAMZAWISRNOJYKMQYECGRSBWHAHLUFBBPDPWLJBRYOCYEAYSVYXSISPRKSNZYPHMM
   WKHXMWWMGAZNEOFMDHKORMGOKNUHTAZQRAZPWBRTQXGZFMTJAXUTRNWCAPZLUFRODLFYFL
   GUKHRODLTYRGRYWHNLRIUCNMDXOCGAKIFAQXKUQMVGZFDBVLSIJSGADLWCFGNCFMGTMWWI
   STBIMHGKXBSPVGFVWHRYHNWXSKNGHLBENHYYQPZLXUEXNHDSBGDQZIXGNQKNUXCCKUFMQI
   MMRYEYUNFHEUDIAZVUJWNGQYSFVSDNZYFNOLWGRBLJGLGTMWWISKZJAXVMXCFVEBMAAHTB
   SNGUPENMWCGBRIFFLHMYOBBBRNZIEHTAZFLTBKMUVGSYVQVMGNZYROHFKISPZLOBBVZHLB
   BKNOYBYRTHVYELSUFXGADJJISBSUTFRPZSGZPTQLQCAZHNGHGADMCCYEEODARGDLSFQHDM
   FIGKZCKYNLDWGHQEDPQHRBSBWLNKDBAMFNOJDSJTFIFMYHZXWXZHQYLBNGSQAWRHMWWQNK
   HMVYPEZLWXUXVCDFAHSQSMGXOLWWVHTMLCZXHHOUVMHHYZBKQYAHSHQWWGRGSMFIEPHFDB
   RMTLFBVLZLESOTBEXIEYQYKBFNOJDCRLAOLWEHRMWMGADYFYZRRZJIAMHYJQVMGIMNQXKU
   QNUXUUDORHENAGRMGULCFUDCFANEHNLFRTGYSXBYXIMLBIOIFYAMGUKWBNMNWXSHQGGLRM
   GUFYVMGYJHHFDLAWNEROHYEBNLANLHQNZYABBYKNPTKWMFNMHIFMJBSBJYTTQXLIPHLGAM
   FTQCSN
   ```

Assume that encrypting with the key letter `A` results in no change, `B` results in an increment by one place in the alphabet, `C` results in an increment by two places, and so on.

What is the key? (Show your work.)

2. Here is a Python dictionary of the relative frequency of letters in English text:

```
{ "A": .08167, "B": .01492, "C": .02782, "D": .04253, "E": .12702, "F": .02228,
  "G": .02015, "H": .06094, "I": .06966, "J": .00153, "K": .00772, "L": .04025,
  "M": .02406, "N": .06749, "O": .07507, "P": .01929, "Q": .00095, "R": .05987,
  "S": .06327, "T": .09056, "U": .02758, "V": .00978, "W": .02360, "X": .00150,
  "Y": .01974, "Z": .00074 }
```

Here is some plaintext:

```
ethicslawanduniversitypolicieswarningtodefendasystemyouneedtobeabletot
hinklikeanattackerandthatincludesunderstandingtechniquesthatcanbeusedt
ocompromisesecurityhoweverusingthosetechniquesintherealworldmayviolate
thelawortheuniversitysrulesanditmaybeunethicalundersomecircumstancesev
enprobingforweaknessesmayresultinseverepenaltiesuptoandincludingexpuls
ioncivilfinesandjailtimeourpolicyineecsisthatyoumustrespecttheprivacya
ndpropertyrightsofothersatalltimesorelseyouwillfailthecourseactinglawf
ullyandethicallyisyourresponsibilitycarefullyreadthecomputerfraudandab
useactcfaaafederalstatutethatbroadlycriminalizescomputerintrusionthisi
soneofseverallawsthatgovernhackingunderstandwhatthelawprohibitsifindou
btwecanreferyoutoanattorneypleasereviewitsspoliciesonresponsibleuseoft
echnologyresourcesandcaenspolicydocumentsforguidelinesconcerningproper
```

The *population variance* of a finite population X of size N and mean μ is given by

$$\mathrm{Var}(\mathrm{X}) = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} (\mathrm{x_i} - \mu)^2.$$

(a) **(5 points + 5 points)** What is the population variance of the relative letter frequencies in English text? What is the population variance of the relative letter frequencies in the given plaintext?

(b) **(10 points)** For each of the following keys — `yz`, `xyz`, `wxyz`, `vwxyz`, `uvwxyz` — encrypt the plaintext with a Vigenère cipher and the given key, then calculate and report the population variance of the relative letter frequencies in the resulting ciphertext. Describe and briefly explain the trend in this sequence of variances.

(c) **(10 points)** Viewing a Vigenère key of length k as a collection of k independent Caesar ciphers, calculate the mean of the frequency variances of the ciphertext for each one. (E.g., for key `yz`, calculate the frequency variance of the even numbered ciphertext characters and the frequency variance of the odd numbered ciphertext characters. Then take their mean.) Report the result for each key in part (c). Is the mean variance like those observed in part (b)? Part (c)? Briefly explain.

(d) **(10 points)** Consider the ciphertext that was produced with key `uvwxyz`. In part (d), you calculated the mean of six variances for this key. Revisit that ciphertext, and calculate the mean of the frequency variances that arise if you had assumed that the key had length 2, 3, 4, and 5. Does this suggest a variant to the Kasiski attack? (Don't say no!) Briefly explain.

3. **(10 points)** Are all the 56 bits of the DES key used an equal number of times in the K_i ? Specify, for each of the K_i , which bits are not used. Here, K_i is the round key for round i.

4. **(10 points)** What is the remainder when 3^{287} is divided by 23? Show your work. Hints: Use the modular arithmetic properties that we discussed in the class.

5. **(6 points + 4 points)** Given $p = 71$, $q = 97$, **find** $\phi(n)$. If $e = 197$, **find** d. Accordingly, given message $m = 8720$, **find the encrypted message** and **decrypted message**. Show your work. Show how the factorization of n breaks RSA.